



Ada Apa dengan Android?

Android merupakan salah satu sistem operasi smartphone ‘junior’ yang baru ‘lahir’ beberapa tahun belakangan ini. Namun, kehadirannya tidak dapat dianggap sebelah mata karena pengguna sistem operasi Android saat ini telah mengalami peningkatan yang luar biasa. Hal ini diakibatkan dengan dukungan vendor smartphone yang mengadopsi sistem operasi baru ini dalam berbagai produk mereka. Sehingga secara tidak langsung pengguna akan menggunakan sistem operasi Android ketika membeli gadget tersebut.

Mungkin ada di antara pembaca yang masih bingung, ketika mendengar istilah Android. Apakah itu semacam sistem operasi atau bahasa pemrograman?

1.1 Sistem Operasi atau Bahasa Program?

Secara sederhana, Android merupakan software yang digunakan pada perangkat mobile yang mencakup sistem operasi, middleware, dan aplikasi kunci yang dirilis oleh Google. Sehingga Android mencakup keseluruhan sebuah aplikasi, mulai dari sistem operasi sampai pada pengembangan aplikasi itu sendiri. Dan pengembangan aplikasi pada platform Android ini menggunakan dasar bahasa pemrograman Java.

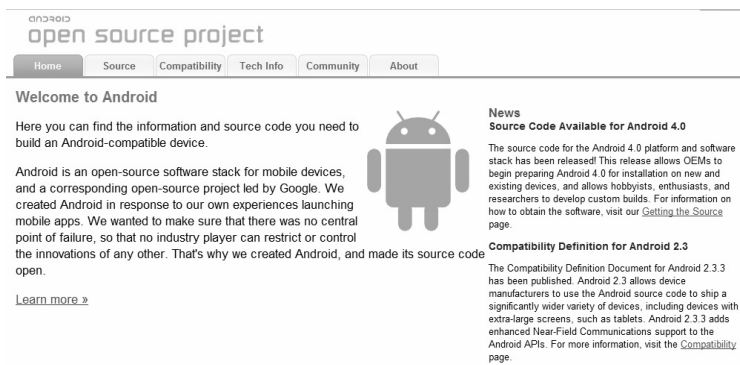
Platform pengembangan aplikasi Android ini bersifat *open-source* atau terbuka sehingga Anda dapat mengembangkan kemampuan untuk membangun aplikasi yang kaya dan inovatif. Bahkan seorang pengembang atau developer Android dapat membuat aplikasi yang bervariasi kemudian menjual untuk keuntungan pribadi tanpa ada lisensi ke produsen atau vendor tertentu.



Gambar 1.1 Logo Android

Android sebagai sistem operasi yang bersifat *open-source*, sama ketika Anda menggunakan sistem operasi Linux dengan berbagai varian distro-nya. Anda dapat mengunduh secara gratis sistem operasi Android, jika ingin mengetahui bagaimana sebenarnya kode program yang digunakan. Atau bahkan lebih daripada itu, Anda ingin memodifikasi sistem operasi Android tersebut. Dan hal itu legal serta diperbolehkan.

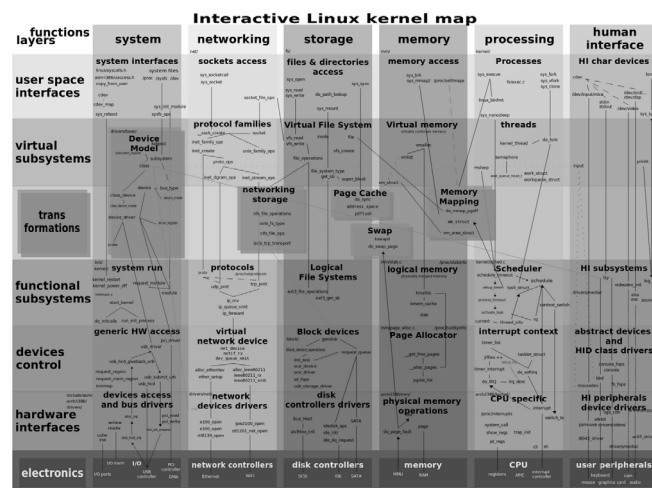
Anda dapat mengakses pada halaman repository Android dan mendapatkan source code Android, pada alamat <http://source.android.com/>.



Gambar 1.2 Repository source code Android

Android dibuat dengan dasar kernel Linux 2.6. Tim pengembang Android memilih kernel Linux tersebut dengan beberapa alasan, antara lain:

- Security, kernel Linux ini dapat mengatur keamanan antara sistem dan aplikasi.
- Manajemen memori, kernel Linux ini dapat mengatur manajemen memori sehingga lebih hemat ketika melakukan develop aplikasi.
- Manajemen proses, kernel Linux ini dapat mengatur proses lebih baik, mengatur *resource* yang diperlukan sesuai dengan kebutuhan untuk menjalankan aplikasi.
- Network stack, kernel Linux ini dapat mengatur komunikasi jaringan.
- Driver, kernel Linux ini menjamin sesuatu dapat berjalan dengan baik, berbagai manufaktur hardware akan dapat bekerja dengan kernel ini.

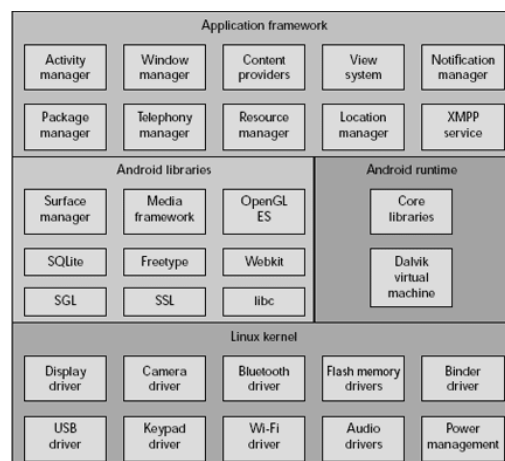


Gambar 1.3 Linux Kernel

Berdasarkan kernel Linux 2.6 tersebut maka framework atau kerangka kerja sistem operasi Android dikembangkan dengan berbagai variasi fiturnya. Fitur-fitur ini kemudian juga dikembangkan secara open-source sehingga Anda pun dapat menambahkan atau memodifikasi sendiri. Berbagai fitur yang ada, antara lain:

- Android run-time, terdiri atas library Java dan Dalvik *virtual machine*.

- Open GL (graphics library), merupakan API (*Application Program Interface*) yang digunakan untuk membuat grafis 2D dan 3D.
- WebKit, merupakan *engine* web browser yang dapat digunakan untuk menampilkan isi website dan menyederhanakan tampilan proses loading.
- SQLite, merupakan *engine* dari relasional database yang dapat diintegrasikan dengan aplikasi.
- Media framework, merupakan library yang digunakan untuk menjalankan dan merekam file audio atau video.
- SSL (*Secure Socket Layer*), merupakan library yang digunakan untuk keamanan internet (*internet security*).



Gambar 1.4 Android Library

Kemudian, bagaimana berbagai library tersebut seperti SQLite atau surface manager berinteraksi dengan Android dalam proses pembuatan aplikasi sendiri. Hal tersebut sudah diantisipasi oleh para pengembang Android sehingga semua library tersebut sudah dapat digunakan secara otomatis ketika mengembangkan aplikasi. Berbagai fitur-fitur yang ada pada library yang dapat digunakan secara langsung, antara lain:

- Activity Manager, digunakan untuk manajemen activity life cycle.

- Telephony Manager, digunakan untuk akses layanan telepon juga informasi daftar kontak, seperti nomor telepon, alamat, dan lain sebagainya.
- View System, digunakan untuk meng-*handle* view dan layout tampilan UI (*user interface*).
- Location Manager, digunakan untuk menandai lokasi geografis pemegang *device*.



Gambar 1.5 Application Framework

Dari kernel sampai develop aplikasi, telah digunakan teknologi yang bersifat open-source sehingga memudahkan seorang developer mengembangkan aplikasi yang kaya fitur. Berikut gambaran keseluruhan kesimpulan dari Android application framework.



Gambar 1.6 Android Application Framework

Perkembangan sistem operasi Android terlihat seperti pada tabel berikut ini.

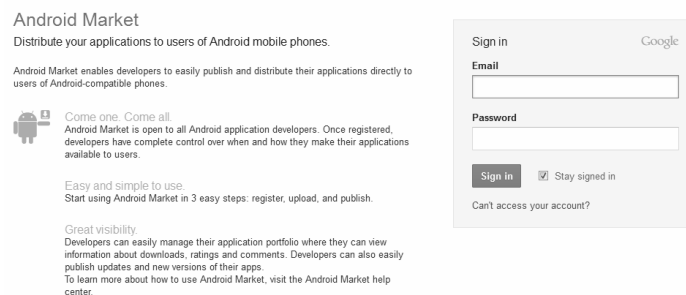
Versi	Tanggal Rilis	Nama
Android 1.0	23 September 2008	
Android 1.1	9 Februari 2009	
Android 1.5	30 April 2009	Cupcake
Android 1.6	15 September 2009	Donut
Android 2.0	26 Oktober 2009	Eclair
Android 2.0.1	3 Desember 2009	Eclair
Android 2.1	12 Januari 2010	Eclair
Android 2.2	20 Mei 2010	Froyo (frozen yoghurt)
Android 2.3	6 Desember 2010	Gingerbread
Android 2.3.3	22 Februari 2011	Gingerbread
Android 3.0	22 Februari 2011	Honeycomb (untuk tablet)
Android 3.1	10 Mei 2011	Honeycomb
Android 3.2	18 Juli 2011	Honeycomb
Android 4.0	19 Oktober 2011	Ice Cream Sandwich

1.2 Mengapa Android?

Kenapa? Barangkali pertanyaan itu yang muncul di benak Anda. Bukankah sudah banyak bahasa pemrograman lainnya dan berjalan secara *cross platform* atau berbagai macam platform. Ada beberapa alasan, kenapa kita perlu mempelajari bahasa pemrograman Android, antara lain:

1. Sebagai seorang developer, Anda mempunyai peluang untuk mengembangkan aplikasi yang nantinya akan prospek ke depan di dunia industri. Dengan pengguna gadget Android yang semakin banyak, tentu akan memerlukan aplikasi yang relevan dan sesuai dengan kebutuhan pengguna.
2. Dengan berbagai API yang telah tersedia secara lengkap dan siap pakai, menjadikan proses develop aplikasi berbasis Android akan lebih mudah. Tempat dan waktu menjual aplikasi tersebut (jika Anda ingin mendapatkan uang hasil develop aplikasi) juga jelas dan mudah. Google sudah menyediakan Android Market,

yang berfungsi seperti sebuah 'pasar' pada umumnya, untuk melakukan transaksi jual beli.



Gambar 1.7 Android Market untuk memublikasikan aplikasi

3. Open platform.

Artinya bahwa Android tidak diikat dengan manufaktur hardware tertentu atau vendor tertentu saja. Sehingga semua vendor dapat men-download, memodifikasi, dan mengembangkan aplikasi sesuai dengan keinginan.

4. Android dapat berjalan pada ukuran dan resolusi layar yang berbeda. Selain itu, juga dilengkapi dengan tool pengembangan yang dapat membuat aplikasi dapat cross-compatibility.

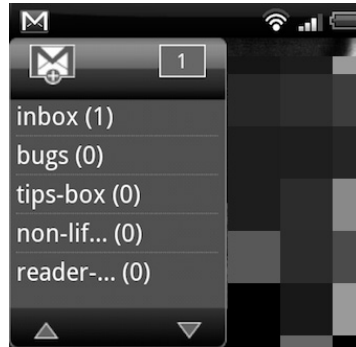
Sebagai sistem operasi, Android juga telah banyak digunakan oleh berbagai vendor ponsel. Ada beberapa kelebihan dari penggunaan ponsel bersistem operasi Android, antara lain:

1. Multitasking.

Sehingga kita bisa menjalankan beberapa aplikasi secara bersamaan. Seperti kita dapat facebook-an sambil mendengarkan audio.

2. Terdapat notifikasi ketika ada panggilan/sms.

Ketika ada SMS dan email yang masuk, akan terdapat notifikasi pada home screen di layar ponsel diikuti dengan lampu LED yang berkedip-kedip, sehingga Anda tidak akan ketinggalan ketika ada pesan sms/email yang masuk.



Gambar 1.8 Notifikasi email yang masuk

3. Dukungan ribuan aplikasi terpercaya melalui situs Android Market.

Google telah menyediakan situs bagi para pengguna Android untuk mendapatkan berbagai macam aplikasi yang diperlukan. Bagi para maniak game, Anda bisa mendapatkan berbagai macam game dalam situs tersebut secara gratis.

4. Penggunaan widget pada home screen sehingga akan memudahkan dan mempercepat pengguna ketika akan membuka aplikasi.

Walaupun begitu, juga terdapat beberapa kelemahan yang perlu Anda ketahui, antara lain:

1. Boros pada penggunaan baterai.

Hal ini karena banyaknya fitur Android, seperti 3G, Maps, Latitude, GPS, dan seterusnya sehingga selalu terkoneksi ke internet. Untuk menghemat, Anda bisa mematikan fitur yang dirasa tidak diperlukan.

2. Sentralisasi Google.

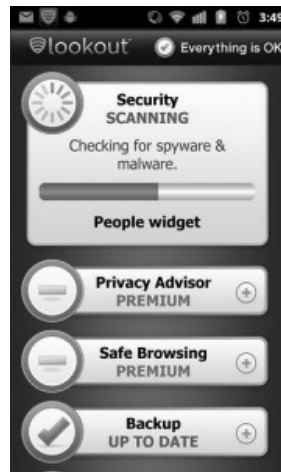
Karena Android merupakan produk *open-source* dari Google maka aplikasi yang berjalan pada Android kebanyakan menyatu dengan layanan Google. Seperti, ketika Anda pertama kali mengaktifkan ponsel Android maka memerlukan registrasi yang menggunakan email di Gmail. Bagi pengguna yang belum terbiasa dengan berbagai produk Google maka harus menyesuaikan diri.

3. Tidak mendukung penggunaan J2ME.

Hal ini berakibat pada seorang programmer Java harus belajar dari awal lagi, dan bagi pengguna, aplikasi yang disukai tidak akan dapat berjalan pada Android. Sebab Android hanya menggunakan bahasanya saja, sedangkan library dan user interface-nya berbeda dengan yang ada di J2ME.

4. Security yang masih tergolong rendah.

Contohnya banyaknya malware yang menyerang aplikasi. Sehingga beberapa produsen Anti Virus menyediakan aplikasi anti virus mereka dalam versi Android, seperti AVG, Lookout Security, Super Security, SmrtGuard Security.



Gambar 1.9 Tampilan Lookout Security

1.3 Dasar Pemrograman Android

Berikut ini beberapa hal dasar atau istilah yang perlu Anda ketahui ketika akan mempelajari bahasa pemrograman Android.

1.3.1 Bahasa Pemrograman Android = Java

Dasar pemrograman Android adalah Java karena aplikasi Android ditulis dalam bahasa Java. Android menyediakan lingkungan atau

run time environment yang dikenal sebagai Dalvik Virtual Machine. Sehingga Dalvik Virtual Machine ini merupakan java runtime environment yang telah dioptimasi untuk device dengan sistem memori yang kecil.

Walaupun begitu, tidak semua murni menggunakan Java, namun masih menggunakan bahasa XML dan dasar Apache Ant untuk pengembangan aplikasi.

Bahasa XML (*eXtensible Markup Language*) merupakan bahasa web turunan dari SGML (Standard Generalized Markup Language) yang ada sebelumnya. XML hampir sama dengan HTML, di mana keduanya diturunkan dari SGML.

Secara sederhana XML adalah suatu bahasa yang digunakan untuk mendeskripsikan dan memanipulasi dokumen secara terstruktur. Secara teknis XML didefinisikan sebagai suatu bahasa meta-markup yang menyediakan format tertentu untuk dokumen-dokumen yang mempunyai data terstruktur. Bahasa Markup adalah mekanisme untuk mengenal suatu struktur di dokumen.

Sedangkan Apache Ant merupakan software berbasis Java yang digunakan untuk keperluan build tool. Sebagai build tool, Apache Ant akan menyediakan sumber daya dan melaksanakan proses yang memungkinkan membangun suatu software dari bentuk source code menjadi aplikasi yang siap didistribusikan.

Build tool tidak diperlukan jika software yang akan kita bangun sangat sederhana (seperti misalnya hanya menampilkan satu atau beberapa baris tulisan saja) dan kita tidak keberatan untuk menuliskan dan mencantumkan berbagai dependencies dari software tersebut (misalnya berbagai file jar yang kita perlukan). Jika aplikasi yang kita bangun relatif kompleks, melibatkan banyak dependencies, dengan struktur direktori tertentu, serta memerlukan banyak eksekusi atau menjalankan tools (misalnya seperti jar, rmic, javac, bzip2, dan lain-lainnya), maka kita memerlukan build tool untuk meng-"otomatiskan" proses pembangunan aplikasi (seperti proses compile, test, run, deploy) software tersebut. Tentu, akan cukup merepotkan ketika kita harus mengetikkan puluhan atau ratusan baris kode untuk melakukan proses siklus compile, test, run, dan deploy secara manual, atau tidak menggunakan build tool.

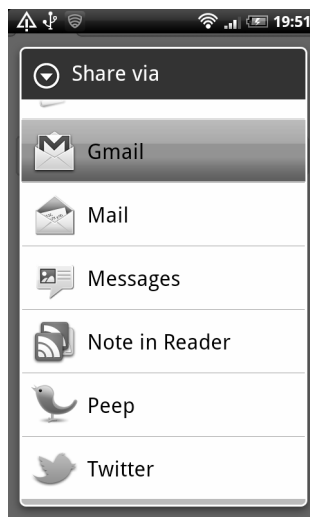
1.3.2 Activity

Sebuah aplikasi Android dibangun di atas satu atau lebih aktivitas. Aplikasi Android minimal terdiri atas satu aktivitas. Bayangkan bahwa activity adalah sebuah kontainer yang terdiri atas rancangan UI (*user-interface*) termasuk kode-kode yang ada di dalamnya.

1.3.3 Intents

Merupakan inti aplikasi Android yang dibangun. Sebuah intent dapat terdiri atas berbagai aksi, seperti view, edit, dial, dan sebagainya. Intent digunakan untuk memulai sebuah activity dan berinteraksi dengan berbagai komponen pada aplikasi Android. Sebuah aplikasi bisa jadi akan mengirim (*broadcast*) atau menerima (*receive*) intent.

Ketika melakukan *broadcast* intent maka akan dikirim pesan ke sistem Android apa yang akan dilakukan sehingga Android akan memulai activity baru atau activity baru pada aplikasi yang lain. Jika semua penerima dapat menerima intent yang dikirim tersebut maka akan muncul pilihan menu/aplikasi mana yang akan dijalankan. Contoh nyatanya adalah ketika Anda menekan lama pada sebuah gambar/*image* di layar ponsel Android maka akan tampil *context-menu* yang berisi menu-menu lainnya.



Gambar 1.10 Memilih menu lain dengan chooser

1.3.4 Kontrol Tanpa Kursor

Jika pada PC yang menggunakan mouse, bisa dilakukan proses klik kanan, maka bagaimana dengan Android?

Jika pada Android, dikenal *long-press* atau menekan lama pada ikon/gambar/menu tertentu sehingga akan tampil *context-menu* yang berisi daftar menu yang lain.

1.3.5 View dan Widgets

View merupakan elemen dasar dari UI, dapat diartikan dengan daerah tertentu yang ada pada layar, di mana kita dapat meletakkan komponen dan melakukan event handling di dalamnya. Beberapa view tersebut, antara lain:

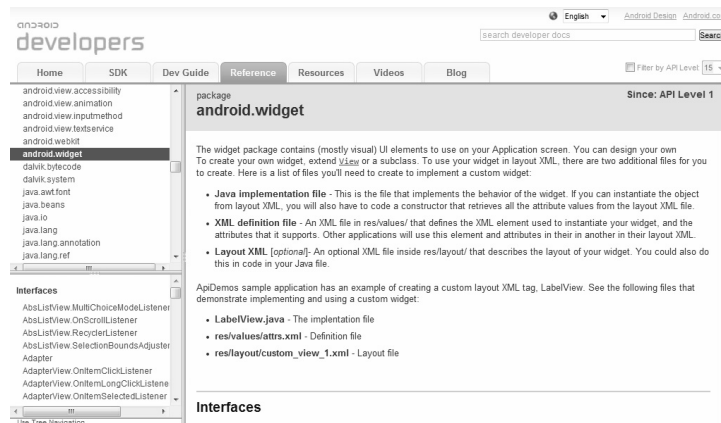
1. Context-Menu
2. Menu
3. View
4. Surface view

Widget merupakan elemen user interface yang berfungsi sebagai antarmuka untuk dapat berinteraksi antara pengguna dengan sistem Android. Beberapa widget yang ada pada Android, antara lain:

1. Checkbox
2. Button
3. Datepicker
4. Digital Clock
5. Gallery
6. Frame layout
7. Image view
8. Relative layout
9. Popupwindow

Sedangkan daftar lengkap widget yang dapat digunakan di Android seperti terlihat pada alamat berikut ini:

<http://developer.android.com/reference/android/widget/package-summary.html>



Gambar 1.11 Halaman penjelasan Android widget

1.3.6 Touchscreen

Fitur yang ini barangkali menjadi salah satu hal yang cukup menarik dan memikat semua pengguna ponsel Android. Dengan fitur ini, Anda dapat menggeser, membalik, memperbesar elemen/semua komponen yang ada pada layar ponsel Android menggunakan sentuhan tangan.

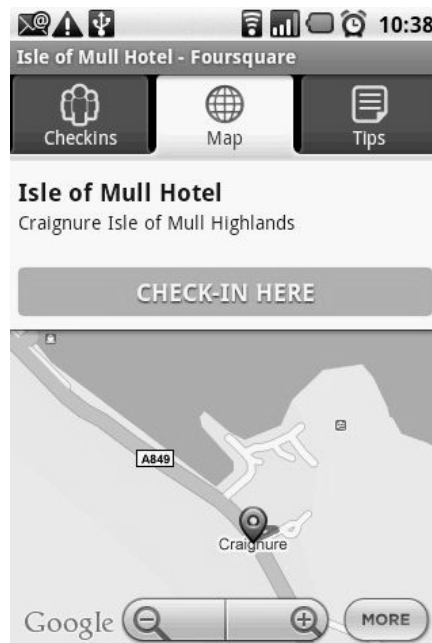
Selain itu, Android juga mendukung fitur *multi-touch* artinya keseluruhan layar dapat disentuh dengan lebih dari satu jari/tangan pada waktu bersamaan.

1.3.7 GPS

Sistem operasi Android dapat dikombinasikan dengan radio GPS sehingga memudahkan developer membuat aplikasi yang dapat mendeteksi lokasi pengguna pada suatu waktu. Selain itu, juga dapat mendeteksi pergerakan atau perubahan lokasi pengguna, ketika berpindah dari lokasi yang semula.

Contoh aplikasi yang ada adalah jaringan sosial four-square yang menggunakan GPS untuk menentukan lokasi ponsel dan mengakses

web untuk menentukan alamat publik yang dekat dengan lokasi pengguna ponsel tersebut.



Gambar 1.12 Aplikasi Foursquare

Atau aplikasi lainnya, yaitu Map atau *peta*, yang dapat menentukan posisi lokasi yang diinginkan kemudian membuat sebuah *roadmap* bagaimana solusi jalan menuju ke posisi tersebut.

1.3.8 SD Card

Android juga dapat berfungsi sebagai tool untuk mengakses sumber penyimpanan dari luar, mengambil, dan menyimpan file dalam *SD card*, misalnya. Dan aplikasi Android yang dibuat dapat pula disimpan pada *SD card*, hal ini akan mengurangi *space* dan juga beban pada ponsel Android ketika menjalankan aplikasi tersebut.



Gambar 1.13 Contoh sebuah Micro SD Card

1.3.9 Dukungan Audio dan Video

Sistem operasi Android dan berbagai aplikasi yang ada di dalamnya mendukung format audio dan video dalam berbagai format. Berbagai efek suara, video, audio dapat ditambahkan dalam aplikasi sesuai dengan keinginan dan kebutuhan.

1.3.10 Kontak

Aplikasi yang Anda buat dapat digunakan untuk mengakses daftar kontak yang tersimpan dalam memori ponsel. Jika Anda bosan dengan aplikasi Kontak yang ada secara default pada ponsel Android, Anda bisa membuat sendiri sesuai keinginan.

1.3.11 Google API

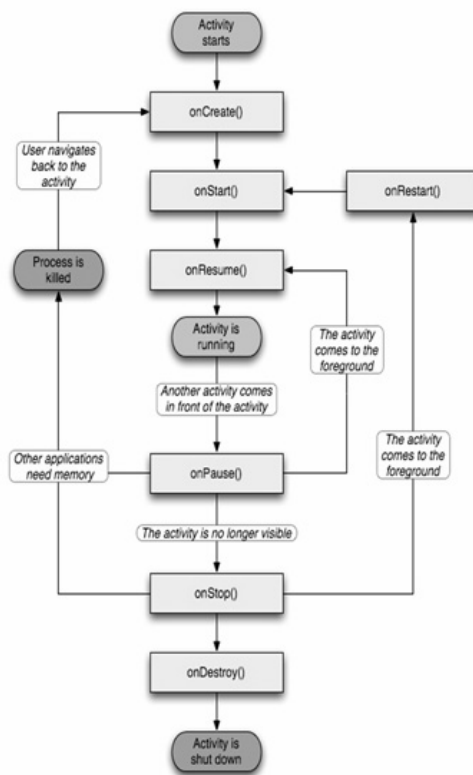
Aplikasi Android yang dibuat dapat diintegrasikan dengan Google API, misalnya seperti pembuatan aplikasi yang digunakan untuk mendeteksi lokasi seorang pengguna. Anda akan memerlukan waktu berjam-jam atau berapa hari dengan ratusan/ribuan kode untuk membuat sebuah *mapping system*. Namun, Google telah menyediakan Android Map API yang dapat digunakan secara langsung dalam aplikasi. Fitur ini tentu akan menghemat waktu serta pikiran Anda dibandingkan harus membuat dari awal. Dan masih banyak Google API lain yang dapat digunakan secara langsung.

1.4 Komponen Android

Sebuah aplikasi Android akan terdiri atas beberapa komponen penting, yaitu:

Activity, merupakan layar tampilan pada sebuah aplikasi Android. Secara sederhananya, activity adalah layar di mana pengguna dapat melihatnya. Sebuah aplikasi Android dapat terdiri atas beberapa activity yang dapat dipindah dalam rentang waktu tertentu ketika menjalankan aplikasi. Activity ini merupakan komponen aplikasi yang umum dan sering berhubungan dengan tampilan, di mana user akan berinteraksi dengan aplikasi yang kita buat.

Skema lifecycle dari sebuah Activity, dapat dilihat pada gambar berikut ini.



Gambar 1.14 Lifecycle Activity

Gambar di atas menunjukkan skema lifecycle sebuah Activity. Ketika sebuah activity dipanggil, kode yang ada pada method onCreate() akan dijalankan, dan method ini juga akan memanggil satu parameter informasi status terakhir yang telah disimpan oleh method onSaveInstanceState().

Method onStart() digunakan untuk mengeksekusi activity ketika sebuah user interface ditampilkan.

Kemudian method lain yang penting, yaitu onPause(), yang akan dipanggil ketika ada activity lain yang akan menggantikan activity yang ada.

Method onStop(), digunakan ketika aplikasi kita sudah tidak dijalankan atau tidak dibutuhkan untuk sementara waktu.

Sedangkan method onRestart(), digunakan ketika activity di-restart dari posisi semula onStop(). Dan method onDestroy(), akan dijalankan setelah activity aplikasi kita di-destroy/ketika habis memori sehingga aplikasi akan di-*terminate*. Secara lengkap, perubahan state activity pada sebuah kode program akan terlihat seperti berikut.

Contohnya:

```
public class ContohActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // activity sedang dibuat.
    }

    @Override
    protected void onStart() {
        super.onStart();
        // Activity mulai terlihat.
    }

    @Override protected void onResume() {
        super.onResume();
        // Activity mulai terlihat, dan di-resume
    }

    @Override protected void onPause() {
        super.onPause();
        // Activity lain mulai fokus, maka activity ini
        akan di-pause.
    }

    @Override protected void onStop() {
        super.onStop();
        // Activity ini mulai tidak ada lagi, sudah
        distop.)
    }
}
```

```

    }

    @Override protected void onDestroy() {
        super.onDestroy();
        // activity ini akan dihancurkan/destroy.
    }
}

```

Views, merupakan widget interface atau antarmuka pada tampilan aplikasi Android, misalnya seperti button, input text. Class yang digunakan dalam Views adalah `android.view.View`. Dan layout dari Views ini diatur dalam subclass yang bernama `android.view.ViewGroups`. Views akan mempunyai atribut yang dapat diubah kelakukannya (diklik, double klik) atau tampilannya (warna, ukuran).

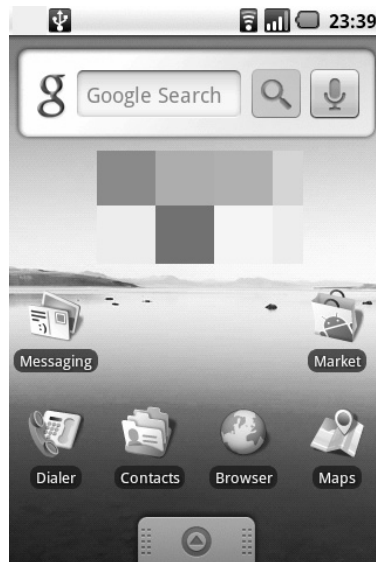
Services, layanan yang dilakukan dari perintah yang diberikan. Dan dijalankan di belakang layar atau tanpa menggunakan *user interface* pengguna. Misalnya seperti MP3 player, yang menjalankan musik selagi pengguna melakukan berbagai activity, seperti browsing, sms, dan sebagainya.

Content Provider, merupakan interface terstruktur untuk data. Hal ini berguna ketika Anda membuat aplikasi Android yang berhubungan dengan penyimpanan dan sharing data ke aplikasi lain.

Intents, merupakan pesan yang bersifat asinkron yang akan memerintahkan aplikasi meminta sesuatu dari komponen yang ada pada sistem Android, misal dari services atau activity. Secara sederhananya, intents merupakan aksi yang akan dilakukan setelah mendapatkan perintah dari services atau activity.

Broadcast Receiver, digunakan untuk menerima pesan dari sistem atau intents.

Widgets, merupakan komponen interaktif yang ada pada layar utama Android. Beberapa ikon widget interaktif akan berada pada tampilan layar yang memperbolehkan pengguna melakukan aksi tertentu. Misalnya ketika akan melihat isi email maka pengguna memilih pada ikon email dan memilih email yang dipilih untuk melihat isi detailnya.



Gambar 1.15 Berbagai widget pada layar Android

Ok, Anda sudah mengetahui bagaimana sebuah sistem operasi Android dan komponen yang ada pada aplikasi Android. Selanjutnya, kita akan belajar membuat aplikasi dengan platform sistem operasi Android.





Persiapkan Amunisi

Semua software atau tool yang digunakan untuk develop aplikasi Android, semuanya bersifat *free*. Artinya, Anda dapat memperoleh berbagai tool pengembangan tersebut secara gratis, tanpa perlu membayar sepeser pun. Hal ini menjadi salah satu keuntungan ketika kita develop aplikasi yang bersifat open-source maka semua tool pengembangnya *free-of-charge* atau gratis, tidak terkecuali pada aplikasi Android juga.

2.1 Sistem Operasi

Spesifikasi sistem operasi yang minimal diperlukan untuk pengembangan aplikasi berbasis Android adalah:

1. Windows XP (32-bit), Vista (32- or 64-bit), atau Windows 7 (32-or 64-bit).
2. Mac OS X 10.5.8 atau versi di atasnya (hanya x86).
3. Linux (testing pada Ubuntu Linux, Lucid Lynx).



Gambar 2.1 Berbagai logo sistem operasi yang didukung Android SDK

2.2 Software

Beberapa software yang diperlukan dalam pengembangan aplikasi Android, yaitu:

1. Java JDK: dasar dari Android SDK.
2. Android SDK: tool untuk mengakses library Android dan menggunakannya untuk develop aplikasi Android.
3. Eclipse IDE (*Integrated Development Environment*): tool untuk menulis kode program Android, juga sebagai tool yang menyatukan antara Java, Android SDK, dan Android ADT.
4. Android ADT: tool untuk membuat file dan struktur yang diperlukan ketika develop aplikasi Android.

2.3 Instalasi dan Konfigurasi

Setelah Anda mengetahui dan mempersiapkan dari sisi hardware atau sistem operasi yang minimal diperlukan ketika akan membangun aplikasi Android, selanjutnya akan dijelaskan langkah-langkah untuk mendapatkan, instalasi, dan konfigurasi tool software pengembang.

2.3.1 Java Development Kit

Langkah-langkah untuk melakukan instalasi Java Development Kit adalah sebagai berikut.

Buka alamat untuk download JDK pada alamat berikut: <http://www.oracle.com/technetwork/java/javase/downloads/index.html>.

Akan terbuka halaman download Java SE seperti gambar berikut, kemudian klik pada link JDK yang ada dalam website tersebut.



Gambar 2.2 Halaman download Java SE

Perhatian:

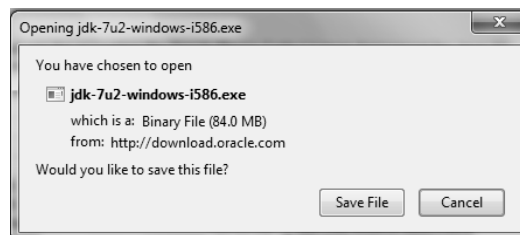
Kadangkala tampilan halaman download JDK akan bisa berbeda. Untuk memastikan bahwa Anda mengakses halaman yang benar, bisa melihat terlebih dahulu pada alamat <http://developer.android.com/sdk/index.html>, kemudian ikuti link download JDK yang dimaksud, sehingga Anda tidak akan salah dalam men-download JDK yang diperlukan.

1. Kemudian pilih sesuai dengan sistem operasi yang Anda gunakan. Ketika buku ini ditulis, versi terakhir adalah versi 7u2. Ada berbagai versi 7u2 untuk platform yang berbeda-beda.



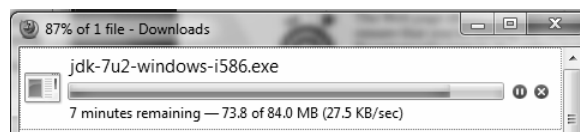
Gambar 2.3 Memilih link sesuai dengan sistem operasi yang digunakan

- Centang pada bagian **Accept License Agreement** kemudian klik link yang dimaksud sehingga akan tampil kotak konfirmasi untuk men-download file JDK, seperti berikut ini.



Gambar 2.4 Pilihan untuk menyimpan file JDK

- Klik tombol **Save File** dan simpan pada folder tertentu di komputer. Tunggu sampai proses download selesai.



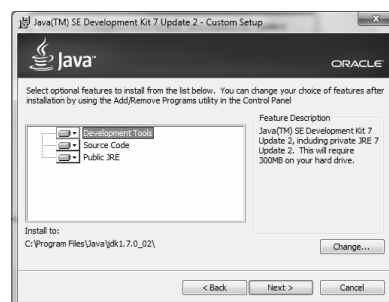
Gambar 2.5 Proses download JDK

- Setelah selesai proses download, klik 2x pada file installer hasil download tersebut hingga tampil langkah instalasi berikut.



Gambar 2.6 Wizard instalasi JDK

5. Kemudian akan tampil kotak dialog pilihan fitur yang akan diinstal dan alamat instalasi. Untuk mengubah alamat lokasi instalasi, klik tombol **Change** dan pilih pada folder mana akan diinstal.



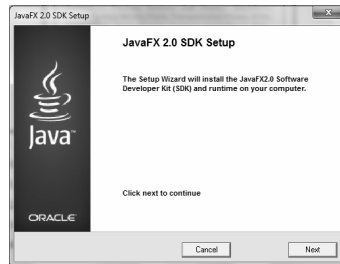
Gambar 2.7 Pemilihan fitur dan alamat instalasi

6. Jika pemilihan fitur dan lokasi sudah selesai, klik tombol **Next**, tunggu sampai proses berjalan.



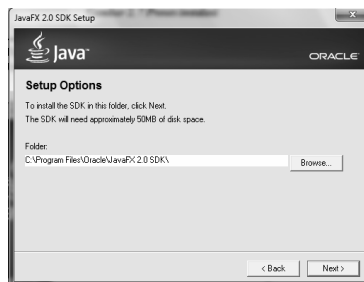
Gambar 2.8 Proses instalasi

7. Tunggu sampai proses selesai, dan muncul kotak wizard instalasi JavaFX 2.0 SDK, seperti berikut.



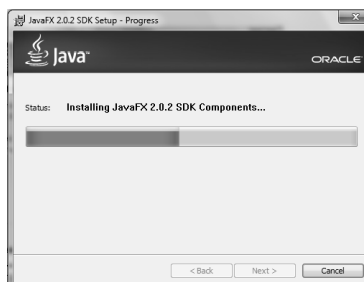
Gambar 2.9 Instalasi JavaFX SDK

8. Klik tombol **Next** untuk melanjutkan proses instalasi selanjutnya. Pilih alamat instalasi yang Anda inginkan. Untuk mengubah lokasi instalasi, klik tombol Browse dan pilih lokasi folder baru untuk instalasi.



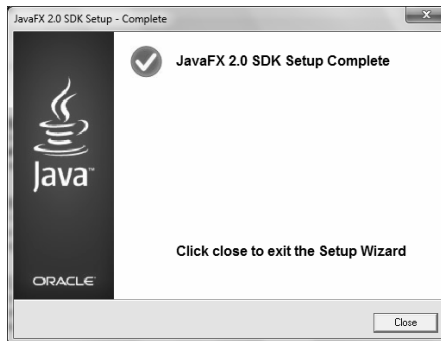
Gambar 2.10 Pemilihan alamat instalasi

9. Klik tombol **Next** sehingga proses instalasi akan berjalan.



Gambar 2.11 Proses instalasi JavaFX SDK

10. Sampai muncul pesan bahwa proses instalasi telah selesai.



Gambar 2.12 Proses instalasi selesai

11. Untuk memastikan bahwa JDK telah terinstal dengan baik, Anda bisa cek menggunakan konsol, dan masukkan perintah `javac -version`, seperti terlihat pada gambar berikut ini.

```
C:\Program Files\Java\jdk1.7.0_02\bin>javac -version
javac 1.7.0_02
C:\Program Files\Java\jdk1.7.0_02\bin>_
```

Gambar 2.13 Cek bahwa JDK telah terinstal

Perhatian:

JavaFX merupakan versi Java yang kaya fitur, yang dapat digunakan untuk membangun atau merancang aplikasi yang kaya dengan konten multimedia, seperti tampilan grafis, suara, efek grafis, dan video. Hal ini menjadi salah satu alasan penulis memilih versi Java yang terbaru ini. Java SDK juga sudah terdapat pula dalam JavaFX ini, sehingga Anda tetap dapat menggunakan versi Java ini untuk develop aplikasi Android.

2.3.2 Android SDK

Untuk melakukan instalasi Android SDK, langkah-langkahnya sebagai berikut:

1. Buka halaman download Android SDK pada alamat <http://developer.android.com/sdk/index.html>. Pada saat buku ini ditulis, versi terakhir adalah r16 seperti gambar berikut.

Download the Android SDK

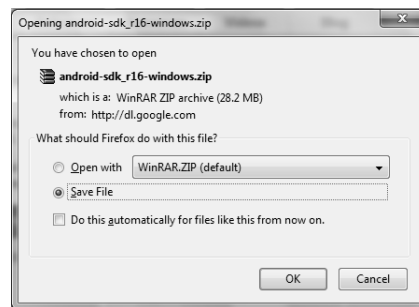
Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK.

If you're already using the Android SDK, you should update to the latest tools or platform using the *Android SDK and AVD Manager*, rather than downloading a new SDK starter package. See [Adding SDK Components](#).

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r16-windows.zip	29562413 bytes	6b926d0c0a871f1a946e65259984701a
	installer_r16-windows.exe (Recommended)	29561554 bytes	3521dda4904886b0598059083c3469
Mac OS X (intel)	android-sdk_r16-macosx.zip	26158334 bytes	d1dc2b6f13ead5e3ce5cf26c4e4c47aa
Linux (i386)	android-sdk_r16-linux.tgz	22048174 bytes	3ba4577731d51da3741c29c8830a4583

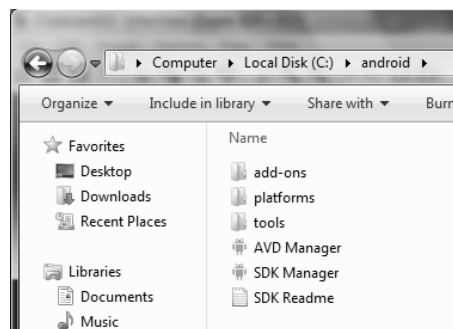
Gambar 2.14 Halaman download Android SDK

2. Kemudian pilih versi terakhir Android SDK yang ada, berikut sesuaikan dengan sistem operasi yang digunakan.



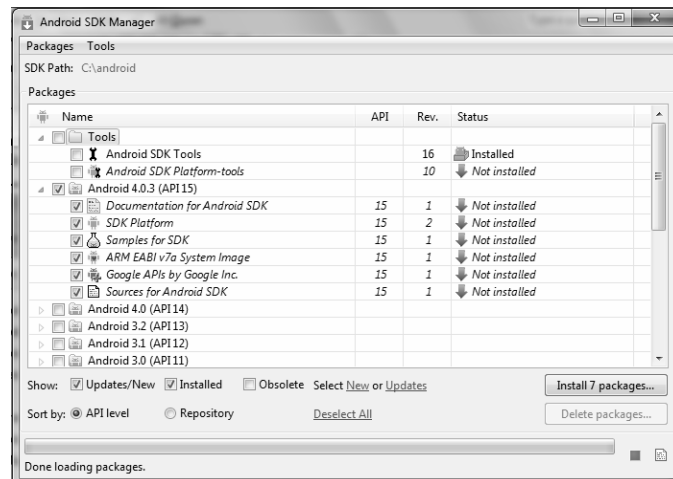
Gambar 2.15 Konfirmasi download Android SDK

3. Simpan pada folder komputer Anda. Kemudian ekstrak hasil download tersebut dalam folder tertentu di komputer.



Gambar 2.16 Hasil ekstrakan Android SDK

4. Klik 2x pada file executable SDK Manager, sehingga akan tampil kotak dialog Android SDK Manager seperti berikut.

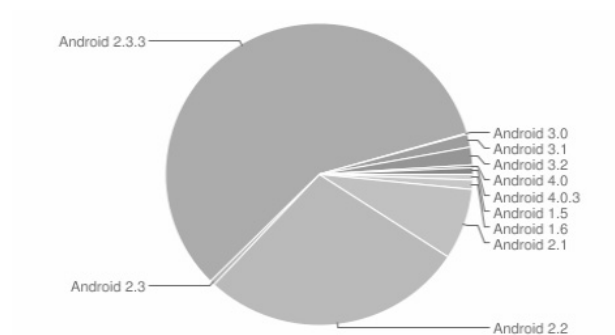


Gambar 2.17 Kotak dialog Android SDK Manager

5. Anda bisa memilih paket-paket yang akan diinstal seperti terlihat di atas. Versi terbaru adalah 4.0.3 (API 15). Tabel berikut menunjukkan versi dan level API dari sistem operasi Android.

Versi	Level API
Android 1.5	API 3
Android 1.6	API 4
Android 2.1	API 7
Android 2.2	API 8
Android 2.3.3	API 10
Android 3.0	API 11
Android 3.1	API 12
Android 3.2	API 13
Android 4.0	API 14
Android 4.0.3	API 15

6. Anda dapat memilih versi yang diinginkan. Namun sebagai pertimbangan, Anda bisa memilih versi Android yang paling banyak digunakan sehingga aplikasi yang Anda buat nantinya ada kemungkinan akan banyak digunakan oleh pengguna. Untuk persentase/prosentase pemakaian berbagai platform/versi Android, statistiknya dapat dilihat pada alamat: <http://developer.android.com/resources/dashboard/platform-versions.html>



Gambar 2.18 Data statistik pengguna OS Android dalam bentuk Piechart

Jika dilihat dalam bentuk tabel, terlihat dengan jelas seperti berikut bahwa versi 2.3.3 (Gingerbread) mempunyai pangsa pasar pengguna yang lebih banyak dibanding versi yang lain, kemudian disusul versi 2.2 (Froyo). Tapi ini tentunya akan selalu berubah seiring dengan perkembangan zaman.

Platform	Codename	API Level	Distribution
Android 1.5	Cupcake	3	0.6%
Android 1.6	Donut	4	1.0%
Android 2.1	Eclair	7	7.6%
Android 2.2	Froyo	8	27.8%
Android 2.3 - Android 2.3.2	Gingerbread	9	0.5%
Android 2.3.3 - Android 2.3.7		10	58.1%
Android 3.0	Honeycomb	11	0.1%
Android 3.1		12	1.4%
Android 3.2		13	1.9%
Android 4.0 - Android 4.0.2	Ice Cream Sandwich	14	0.3%
Android 4.0.3		15	0.7%

Data collected during a 14-day period ending on February 1, 2012

Gambar 2.19 Data statistik pengguna OS Android dalam bentuk tabel

1. Dengan dasar data tersebut, kita akan menggunakan versi 2.3.3 dan ketika tanda minus pada pilihan tersebut diperlebar, terlihat didukung oleh berbagai hardware yang cukup banyak, dibandingkan dengan versi lainnya.

Android 2.3.3 (API 10)			
<input checked="" type="checkbox"/> SDK Platform	10	2	Not installed
<input checked="" type="checkbox"/> Samples for SDK	10	1	Not installed
<input checked="" type="checkbox"/> Google APIs by Google Inc.	10	2	Not installed
<input type="checkbox"/> Dual Screen APIs by KYOCERA Corporation	10	1	Not installed
<input type="checkbox"/> Real3D by LGE	10	1	Not installed
<input type="checkbox"/> ADMIRAL by Motorola Mobility, Inc.	10	5	Not installed
<input type="checkbox"/> ATRIX2 by Motorola Mobility, Inc.	10	1	Not installed
<input type="checkbox"/> Bionic by Motorola Mobility, Inc.	10	2	Not installed
<input type="checkbox"/> Droid4 by Motorola Mobility, Inc.	10	1	Not installed
<input type="checkbox"/> DroidRAZR by Motorola Mobility, Inc.	10	3	Not installed
<input type="checkbox"/> DroidX2 by Motorola Mobility, Inc.	10	1	Not installed
<input type="checkbox"/> MT870 by Motorola Mobility, Inc.	10	2	Not installed

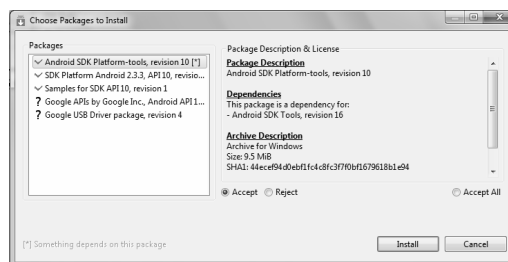
Gambar 2.20 Versi 2.3.3 dengan dukungan bermacam ponsel

2. Pada bagian **Extras**, Anda bisa mencentang di bagian **Google USB Driver Package**. Paket ini berfungsi ketika suatu saat Anda menggunakan device Android sebenarnya sehingga bisa langsung dicolokkan ke komputer.

Extras			
<input type="checkbox"/> Android Support package		6	Not installed
<input type="checkbox"/> Google Admob Ads Sdk package		4	Not installed
<input type="checkbox"/> Google Analytics Sdk package		2	Not installed
<input type="checkbox"/> Google Market Billing package		1	Not installed
<input type="checkbox"/> Google Market Licensing package		1	Not installed
<input checked="" type="checkbox"/> Google USB Driver package		4	Not installed
<input type="checkbox"/> Google Webdriver package		2	Not installed

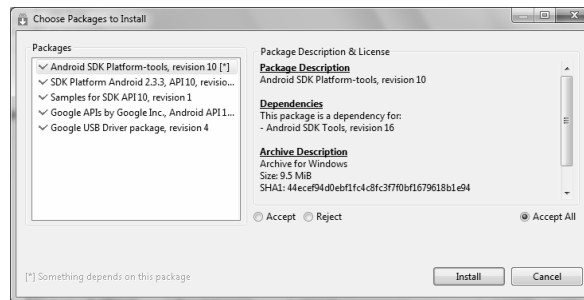
Gambar 2.21 Pilihan Google USB Driver Package

3. Jika sudah selesai memilih paket-paket yang akan diinstal, klik tombol **Install** untuk memulai proses instalasi. Akan tampil kotak dialog **Choose Packages to Install**.



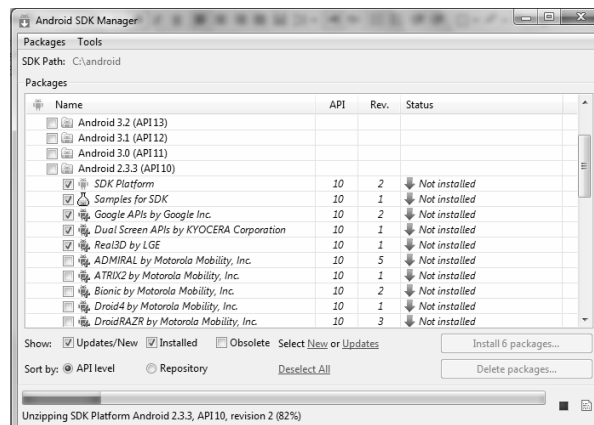
Gambar 2.22 Kotak dialog Choose Packages to Install

4. Klik pada bagian **Accept All** untuk melakukan instalasi semua paket, seperti terlihat pada gambar berikut.



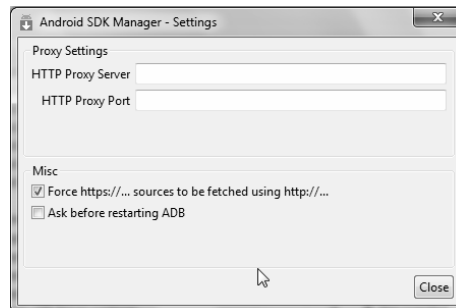
Gambar 2.23 Pilihan untuk instal semua paket

5. Klik tombol **Install** sehingga proses instalasi akan berjalan.



Gambar 2.24 Proses download dan instalasi paket

6. Tunggu sampai proses download selesai. Jika Anda mendapatkan pesan error dan tidak dapat koneksi ke server maka bisa mencoba alternatif lain pada bagian Settings, kemudian pilih pada opsi *Force <https://...sources> to be fetched using http://...*, seperti terlihat pada gambar berikut ini. Coba lakukan proses download lagi.



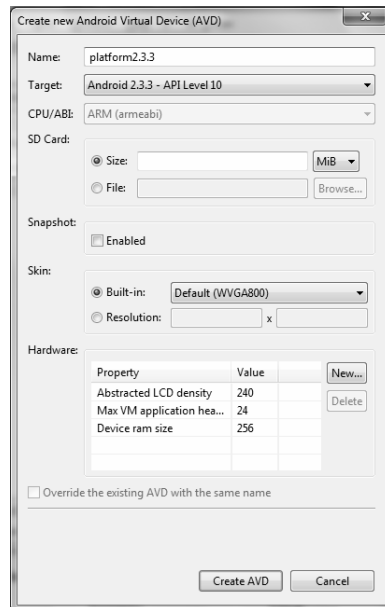
Gambar 2.25 Opsi perubahan nama URL

7. Ketika proses instalasi telah selesai maka status dari paket tersebut akan berubah, dan menunjukkan telah terinstal.

Name	API	Rev.	Status
Android SDK Tools		16	Installed
Android SDK Platform-tools		10	Installed
Android 4.0.3 (API 15)			
Android 4.0 (API 14)			
Android 3.2 (API 13)			
Android 3.1 (API 12)			
Android 3.0 (API 11)			
Android 2.3.3 (API 10)			
SDK Platform	10	2	Installed
Samples for SDK	10	1	Installed
Google APIs by Google Inc.	10	2	Installed
DTS Add-On by KYOCERA Corporation	10	1	Installed
Dual Screen APIs by KYOCERA Corporation	10	1	Not installed
Real3D Add-On by LGE	10	1	Installed

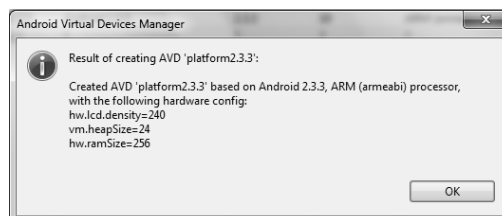
Gambar 2.26 Status paket telah terinstal

8. Kemudian klik AVD Manager, sehingga akan tampil kotak dialog untuk membuat AVD. Setting sesuai dengan versi Android yang akan Anda gunakan.



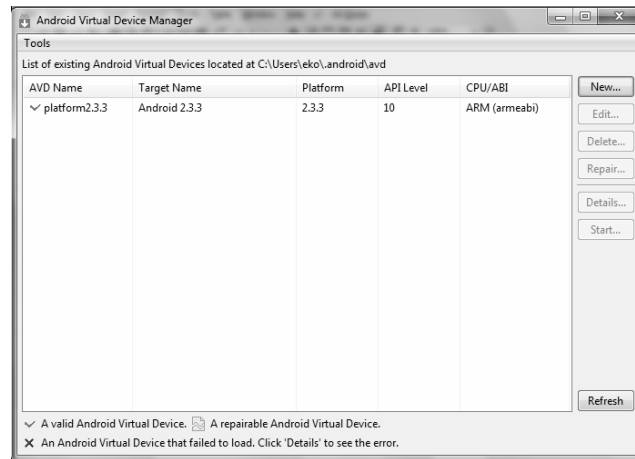
Gambar 2.27 Membuat Android Virtual Device

9. Klik tombol **Create AVD** sehingga akan tampil hasil pembuatan AVD seperti berikut.



Gambar 2.28 Pesan konfirmasi hasil pembuatan AVD

10. Dan pada jendela **Android Virtual Device Manager** akan tampil nama AVD berikut dengan target versi Android yang digunakan.



Gambar 2.29 AVD baru telah terbentuk

Ketika Anda melihat Android SDK hasil ekstrakan, akan terlihat banyak folder di dalamnya. Penjelasan masing-masing folder tersebut adalah sebagai berikut.

- **usb-drivers**, folder yang berisi driver untuk perangkat Android. Jika Anda terkoneksi secara langsung pada perangkat ponsel Android maka akan memerlukan driver sehingga dapat melakukan proses develop secara langsung. Folder ini akan tampil ketika Anda telah dan selesai menginstal USB driver.
- **tools**, folder ini berisi berbagai tool yang berguna ketika proses develop aplikasi, seperti tool untuk debugging, tool untuk tampilan aplikasi (*view-management*), tool untuk develop, dan sebagainya.
- **temp**, folder ini berisi memory swap yang diperlukan untuk space ekstra ketika melakukan proses develop atau menjalankan aplikasi.
- **samples**, folder ini berisi contoh aplikasi yang termasuk kode sumber di dalamnya.
- **platforms**, folder ini berisi informasi platform target yang dipilih ketika develop aplikasi Android, seperti nama folder *android-8* (artinya Android 2.2), *android-4* (artinya Android 1.6), dan lain sebagainya.

- docs, folder ini berisi dokumentasi Android SDK.
- add-ons, folder ini berisi API tambahan yang dapat digunakan ketika proses pembuatan aplikasi.

Perhatian:

Jika Anda akan menggunakan kode program lain seperti C atau C++, Anda bisa menggunakan Android NDK (*Native Development Kit*). Dan Android NDK ini akan melengkapi Android SDK yang telah di-download, dan tidak akan me-*replace* yang telah ada. Anda bisa mendapatkan Android NDK terbaru pada alamat berikut: <http://developer.android.com/sdk/ndk/index.html>.

Download the Android NDK

The Android NDK is a companion tool to the Android SDK that lets you build performance-critical portions of your apps in native code. It provides headers and libraries that allow you to build activities, handle user input, use hardware sensors, access application resources, and more, when programming in C or C++. If you write native code, your applications are still packaged into an .apk file and they still run inside of a virtual machine on the device. The fundamental Android application model does not change.

Using native code does not result in an automatic performance increase, but always increases application complexity. If you have not run into any limitations using the Android framework APIs, you probably do not need the NDK. Read [What is the NDK?](#) for more information about what the NDK offers and whether it will be useful to you.

The NDK is designed for use only in conjunction with the Android SDK. If you have not already installed and setup the [Android SDK](#), please do so before downloading the NDK.

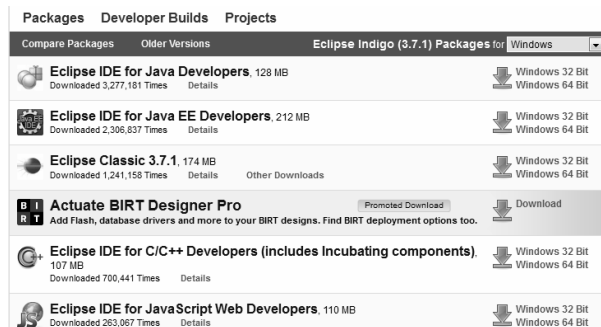
Platform	Package	Size	MD5 Checksum
Windows	android-ndk-r7-windows.zip	81270552 bytes	55483482cf2b75e8dd1a5d9a7caeb6e5
Mac OS X (intel)	android-ndk-r7-darwin-x86.tar.bz2	71262092 bytes	817ca5675a1dd44078098e43070f19b6
Linux 32/64-bit (x86)	android-ndk-r7-linux-x86.tar.bz2	64884365 bytes	bf15e6b47bf50824c4b96849bf003ca3

Gambar 2.30 Halaman download Android NDK

2.3.3 Instalasi Eclipse

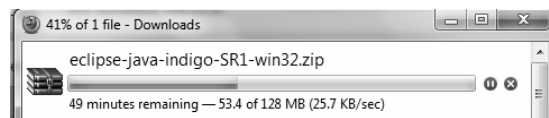
Kemudian kita perlu download dan melakukan instalasi tool editor yang digunakan, yaitu Eclipse. Untuk instalasi Eclipse, langkah-langkahnya sebagai berikut:

1. Download installer Eclipse pada alamat berikut <http://www.eclipse.org/downloads/>. Pilih versi yang terbaru dari daftar Eclipse yang disediakan, pilih pada Eclipse IDE for Java Developer. Anda juga dapat menggunakan Eclipse IDE for Java EE Developers. Sesuaikan dengan sistem operasi yang Anda gunakan, klik menu drop-down pada sistem operasi yang digunakan.



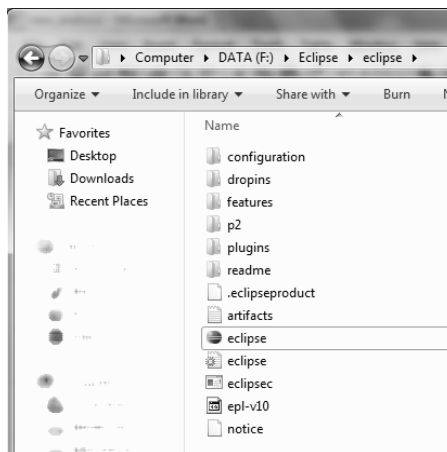
Gambar 2.31 Halaman download Eclipse

2. Penulis menggunakan Eclipse IDE for Java Developers Indigo (versi yang terbaru), klik pada link download, tunggu sampai proses download selesai.



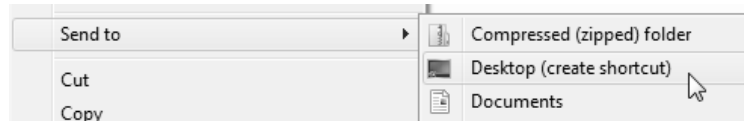
Gambar 2.32 Proses download Eclipse

3. Eclipse merupakan tool editor yang berdiri sendiri, artinya Anda tinggal ekstrak file zip hasil download-an, kemudian klik 2x pada file executable eclipse.



Gambar 2.33 Folder ekstraksi editor Eclipse

4. Anda bisa membuat file shortcut di desktop, dengan tinggal klik kanan pada file exe dan pilih Send to → Desktop (create shortcut).



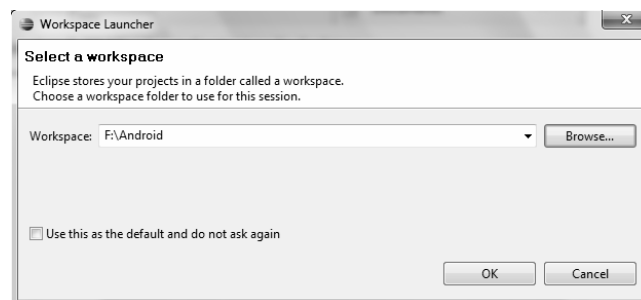
Gambar 2.34 Membuat shortcut pada desktop

5. Klik 2x pada file executable tersebut sehingga akan tampil *screen* tampilan Elipse Indigo, yang menunjukkan versi Eclipse yang akan Anda gunakan.



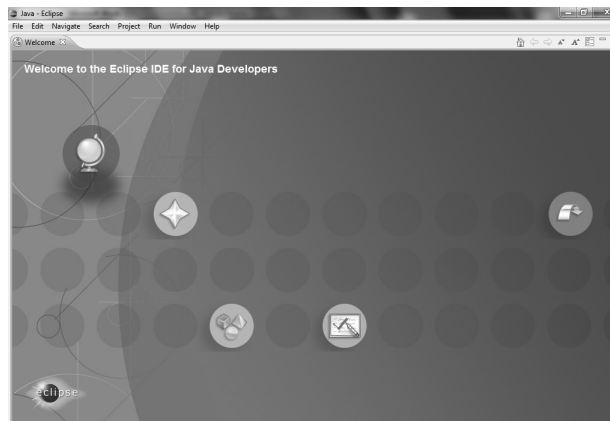
Gambar 2.35 Eclipse Indigo

6. Kemudian akan tampil jendela Workspace launcher seperti berikut. Anda dapat men-setting alamat folder sebagai folder kerja untuk meletakkan file/folder develop aplikasi.



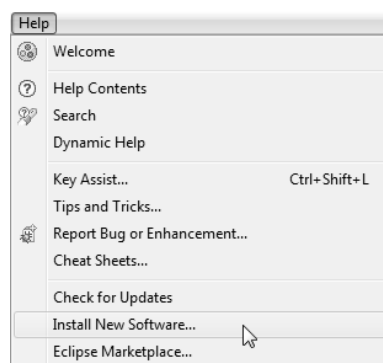
Gambar 2.36 Memilih workspace

7. Akan muncul halaman welcome screen dari Eclipse seperti berikut ini.



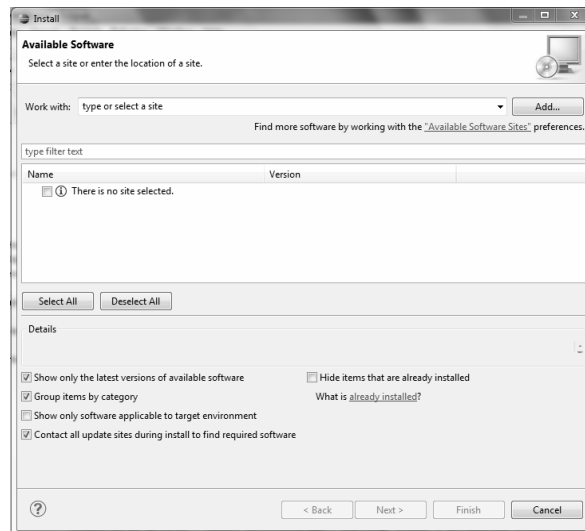
Gambar 2.37 Welcome screen Eclipse

8. Selanjutnya kita akan menggunakan ADT (Android Development Tools) untuk proses pembuatan aplikasi Android. ADT akan mempermudah Anda melakukan proses devel aplikasi Android sehingga proses coding lebih cepat, tanpa harus memulai dari awal. Juga dapat digunakan untuk proses debug aplikasi, deploy aplikasi menjadi bentuk APK (*Android Package*).
9. Untuk mengintegrasikan ADT dalam Eclipse, caranya melalui menu Help → Install New Software.



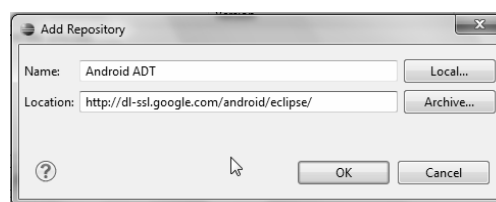
Gambar 2.38 Opsi Install New Software

10. Akan tampil kotak dialog **Install** seperti berikut. Kotak dialog berikut akan memandu Anda menginstal plug-in baru dalam Eclipse.



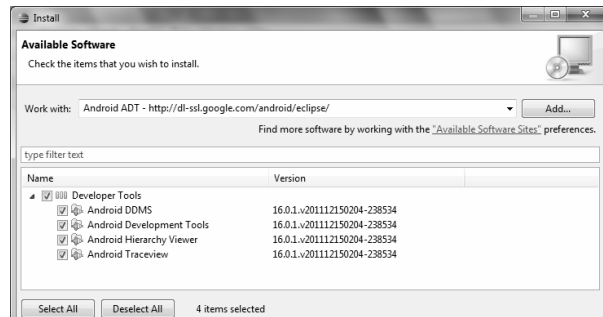
Gambar 2.39 Kotak dialog *Install* untuk menambahkan software baru

11. Klik tombol **Add** pada bagian **Work with**, sehingga tampil kotak dialog **Add repository** seperti berikut ini. Masukkan pada bagian Name dan Location seperti berikut. Misal, Name diisi dengan 'Android ADT' dan Location dengan <http://dl-ssl.google.com/android/eclipse/>.



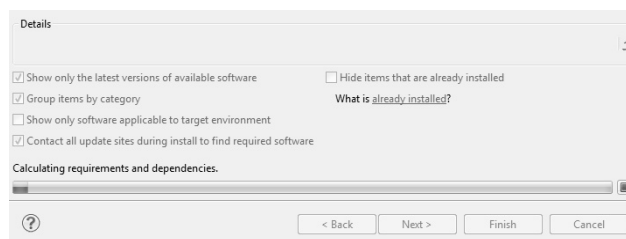
Gambar 2.40 Menambah repository baru

12. Kemudian akan tampil pada jendela di bawahnya, plug-in baru yang akan ditambahkan. Centang pada bagian **Developer Tools** dan pastikan submenu di bawahnya juga ikut tercentang.



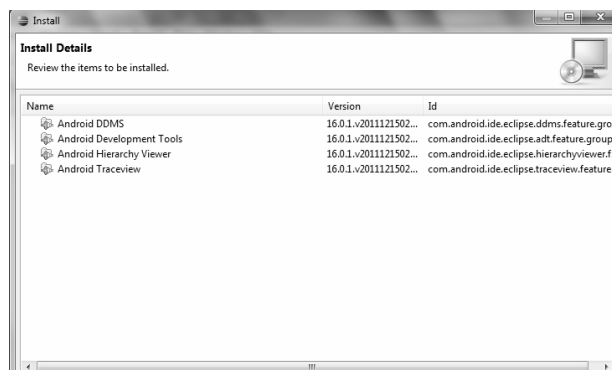
Gambar 2.41 Plug-in yang akan ditambahkan

- Klik tombol **Next**, maka proses pengecekan dan software lain yang diperlukan akan dijalankan. Tunggu sampai proses selesai.



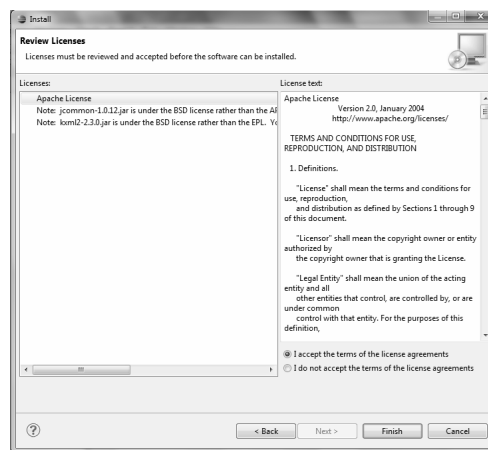
Gambar 2.42 Proses instalasi plug-in tambahan baru

- Kemudian akan tampil review plug-in yang akan diinstal, terlihat seperti gambar berikut ini.



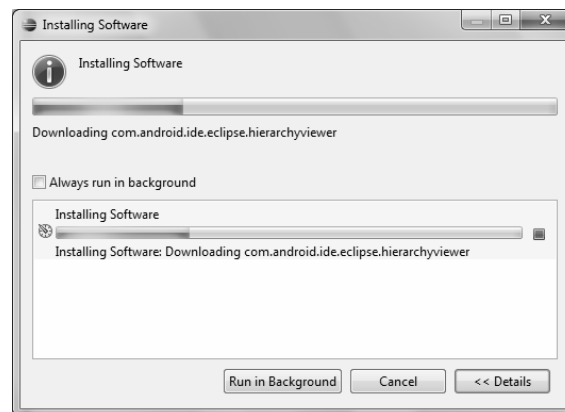
Gambar 2.43 Review plug-in yang akan diinstal

15. Jika ada yang salah atau ada plug-in yang kurang, Anda bisa klik tombol **Back** untuk memilih ulang. Jika sudah benar, klik tombol **Next**, sehingga tampil kotak **Review Licenses** seperti berikut. Klik pada bagian *'I accept the terms of the license agreements'*.



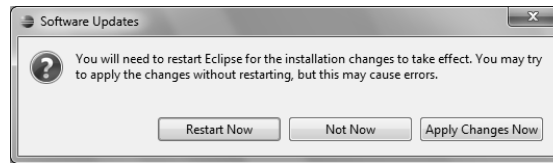
Gambar 2.44 Kotak dialog Review Licenses

17. Klik tombol **Finish** sehingga proses download akan dijalankan.



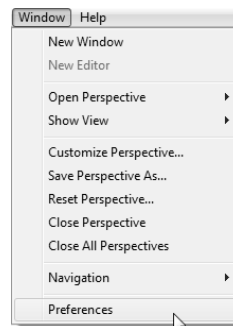
Gambar 2.45 Proses download dan instalasi plug-in baru

18. Tunggu sampai proses selesai dan akan tampil kotak konfirmasi untuk me-restart **Eclipse** yang digunakan.



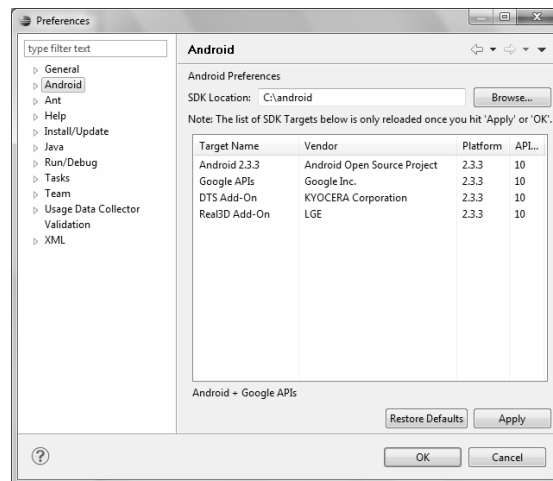
Gambar 2.46 Pilihan untuk restart Eclipse

19. Kemudian kita perlu men-setting lokasi SDK yang akan digunakan. Pilih pada menu **Windows** → **Preferences**, seperti berikut.



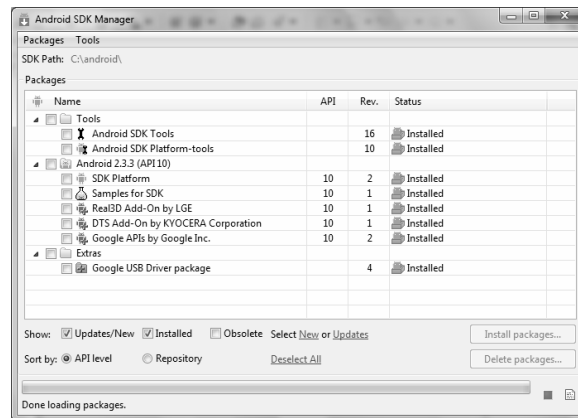
Gambar 2.47 Opsi Preferences

20. Kemudian pilih pada bagian Android dan pilih lokasi SDK pada bagian **SDK Location**, klik tombol **Browse**.



Gambar 2.48 Memilih lokasi SDK yang digunakan

21. Cek SDK yang digunakan, apakah telah benar versi yang digunakan.



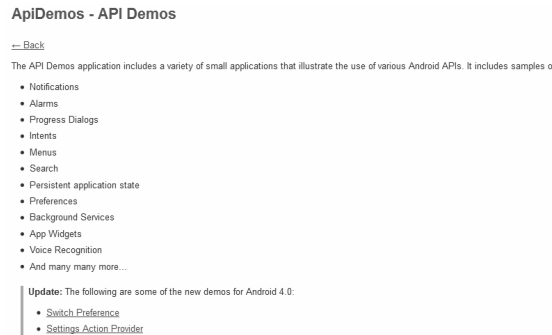
Gambar 2.49 Kotak dialog Android SDK Manager

22. Ok, sampai proses ini Anda sudah melakukan setting semua tool yang diperlukan untuk proses pembuatan aplikasi Android.
23. SDK menyediakan emulator kecil yang digunakan untuk melakukan testing atau uji coba aplikasi. Namun, tidak semua aplikasi dapat di testing menggunakan emulator, ada beberapa aplikasi seperti aplikasi accelerometer (untuk mengukur kecepatan), tentu Anda membutuhkan sebuah alat tambahan atau ponsel Android yang terhubung ke komputer. Sehingga saat kita menginstal USB driver, dan ketika memerlukan peralatan tambahan tetap dapat menggunakannya. Jika Anda menggunakan sistem operasi Mac atau Linux, maka langsung dapat menggunakan tanpa memerlukan driver.

Google APIs by Google Inc.	10	2	Installed
Extras			
Google USB Driver package		4	Installed

Gambar 2.50 Instalasi USB Driver

24. Contoh penggunaan API dalam pengembangan aplikasi berbasis Android, dapat melihat pada halaman: <http://developer.android.com/resources/samples/ApiDemos/index.html>



Gambar 2.51 Halaman contoh penggunaan API

Beberapa aplikasi yang menggunakan API ini, antara lain:

- Notification
- Alarm
- Intents
- Menus
- Search
- Preferences

2.4 Proses Debugging

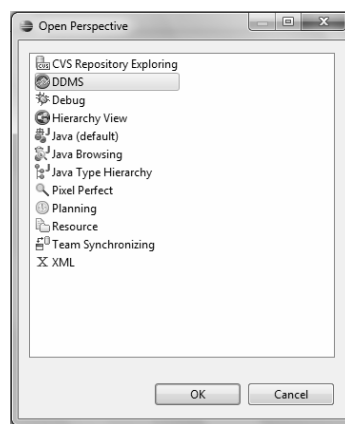
Debugging adalah sebuah metode yang dilakukan oleh para programmer dan pengembang perangkat lunak untuk mencari dan mengurangi *bug* aplikasi yang dibuat, atau kerusakan di dalam sebuah program komputer atau perangkat keras sehingga perangkat tersebut bekerja sesuai dengan harapan.

Debugging cenderung lebih rumit ketika terdapat beberapa subsistem lainnya yang terikat dengannya, karena ketika dilakukan sebuah perubahan di satu sisi, mungkin dapat menyebabkan munculnya *bug* lain di dalam subsistem lainnya. Untuk melakukan proses debug pada aplikasi Android yang Anda develop, dapat menggunakan DDMS (*Dalvik Debug Monitor System*).

Anda dapat melihat tool DDMS ini pada alamat <http://developer.android.com/guide/developing/tools/ddms.html>.

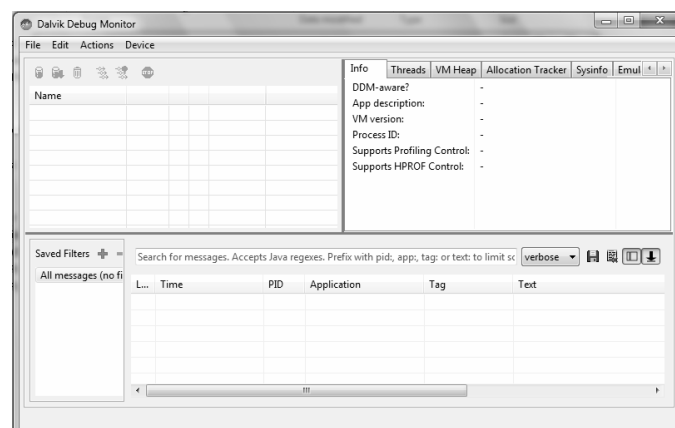
DDMS ini terintegrasi dalam Eclipse dan juga terdapat pada folder *tools/* yang ada dalam Android SDK. DDMS akan dapat bekerja secara simultan dengan emulator dan perangkat ponsel yang terkoneksi.

Pada Eclipse, Anda bisa membukanya pada menu Window → Open Perspective → Other, sehingga akan tampil kotak dialog Open Perspective. Pilih pada bagian DDMS.



Gambar 2.52 Kotak dialog Open Perspective

Jika pada folder *tools/* maka pilih pada file *ddms.bat*. Jalankan file tersebut dengan klik 2x, sehingga akan tampil seperti berikut.



Gambar 2.53 Dalvik Debug Monitor

Ada 5 fungsi penting dari penggunaan DDMS ini, antara lain:

- Untuk manajemen file, dengan DDMS ini kita dapat melihat file-file yang ada pada device ponsel atau file-file pada emulator. Kita juga dapat copy-paste file dari Android ke komputer kita, atau bisa juga sebaliknya.
- Melakukan interaksi dengan emulator, dengan DDMS maka kita dapat mengirimkan events langsung ke emulator seperti melakukan pengiriman sms, melakukan dial/panggilan, dan sebagainya.
- Capture screen/layar, dengan DDMS ini dapat dilakukan untuk meng-capture tampilan pada emulator atau sebuah device.
- Task management, dengan DDMS ini kita dapat menelusuri dengan detail proses yang ada, berikut dengan kontrol untuk menghentikan proses tersebut, baik pada sebuah emulator maupun pada device.
- Logging, dengan DDMS ini kita dapat melihat file-file log pada sebuah emulator.

